

LOG4J - LOGGING IN FILES

http://www.tutorialspoint.com/log4j/log4j_logging_files.htm

Copyright © tutorialspoint.com

To write your logging information into a file you would have to use *org.apache.log4j.FileAppender*. There are following configurable parameters of FileAppender:

FileAppender Configuration:

Property	Description
immediateFlush	This flag is by default set to true, which means the output stream to the file being flushed with each append operation.
encoding	It is possible to use any character-encoding. By default is the platform-specific encoding scheme.
threshold	The threshold level for this appender.
Filename	The name of the log file.
fileAppend	This is by default set to true, which mean the logging information being appended to the end of the same file.
bufferedIO	This flag indicates whether we need buffered writing enabled. By default is set to false.
bufferSize	If bufferedI/O is enabled, this indicates the buffer size. By default is set to 8kb.

Following is a sample configuration file *log4j.properties* for FileAppender.

```
# Define the root logger with appender file
log4j.rootLogger = DEBUG, FILE

# Define the file appender
log4j.appender.FILE=org.apache.log4j.FileAppender
# Set the name of the file
log4j.appender.FILE.File=${log}/log.out

# Set the immediate flush to true (default)
log4j.appender.FILE.ImmediateFlush=true

# Set the threshold to debug mode
log4j.appender.FILE.Threshold=debug

# Set the append to false, overwrite
log4j.appender.FILE.Append=false

# Define the layout for file appender
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.conversionPattern=%m%n
```

If you love to have an XML configuration file equivalent to above *log4j.properties* file then here is the content:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration>

<appender name="FILE" >
```

```

<param name="file" value="${log}/log.out"/>
<param name="immediateFlush" value="true"/>
<param name="threshold" value="debug"/>
<param name="append" value="false"/>
<layout >
    <param name="conversionPattern" value="%m%n"/>
</layout>
</appender>

<logger name="log4j.rootLogger" additivity="false">
    <level value="DEBUG"/>
    <appender-ref ref="FILE"/>
</logger>

</log4j:configuration>

```

You can try [log4j - Sample Program](#) with the above configuration .

Logging in Multiple Files:

There may be a requirement when you want to write your log message into multiple files for certain reasons like for example if file size reaches to a certain threshold etc.

To write your logging information into multiple files you would have to use *org.apache.log4j.RollingFileAppender* class which extends the *FileAppender* class and inherits all its properties.

There are following configurable parameters in addition to what have been mentioned above for FileAppender:

Property	Description
maxFileSize	This is the critical size of the file above which the file will be rolled. Default value is 10MB
maxBackupIndex	This property denotes the number of backup files to be created. Default value is 1.

Following is a sample configuration file *log4j.properties* for RollingFileAppender.

```

# Define the root logger with appender file
log4j.rootLogger = DEBUG, FILE

# Define the file appender
log4j.appender.FILE=org.apache.log4j.RollingFileAppender
# Set the name of the file
log4j.appender.FILE.File=${log}/log.out

# Set the immediate flush to true (default)
log4j.appender.FILE.ImmediateFlush=true

# Set the threshold to debug mode
log4j.appender.FILE.Threshold=debug

# Set the append to false, should not overwrite
log4j.appender.FILE.Append=true

# Set the maximum file size before rollover
log4j.appender.FILE.MaxFileSize=5KB

# Set the the backup index
log4j.appender.FILE.MaxBackupIndex=2

# Define the layout for file appender
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.conversionPattern=%m%n

```

If you like to have an XML configuration file, you can generate as mentioned in initial section and add only additional parameters related to *RollingFileAppender*.

This example configuration demonstrates that maximum permissible size of each log file is 5MB. Upon exceeding the maximum size, a new log file will be created and because *maxBackupIndex* is defined as 2, once the second log file reaches the maximum size, the first log file will be erased and thereafter all the logging information will be rolled back to the first log file.

You can try [log4j - Sample Program](#) with the above configuration .

Daily Log File Generation:

There may be a requirement when you want to generate your log files on per day basis to keep a clean record of your logging information.

To write your logging information into files on daily basis you would have to use *org.apache.log4j.DailyRollingFileAppender* class which extends the *FileAppender* class and inherits all its properties.

There is only one important following configurable parameter in addition to what have been mentioned above for *FileAppender*:

Property	Description
DatePattern	This indicates when to roll over the file, and the naming convention to be followed. By default roll over at midnight each day.

DatePattern controls the rollover schedule using one of the following patterns:

DatePattern	Description
'.' yyyy-MM	Roll over at the end of each month and the beginning of the next month.
'.' yyyy-MM-dd	This is the default value and roll over at midnight each day.
'.' yyyy-MM-dd-a	Roll over at midday and midnight of each day.
'.' yyyy-MM-dd-HH	Roll over at the top of every hour.
'.' yyyy-MM-dd-HH-mm	Roll over every minute.
'.' yyyy-ww	Roll over on the first day of each week depending upon the locale.

Following is a sample configuration file *log4j.properties* to generate log files rolling over at midday and midnight of each day.

```
# Define the root logger with appender file
log4j.rootLogger = DEBUG, FILE

# Define the file appender
log4j.appender.FILE=org.apache.log4j.DailyRollingFileAppender
# Set the name of the file
log4j.appender.FILE.File=${log}/log.out

# Set the immediate flush to true (default)
```

```
log4j.appender.FILE.ImmediateFlush=true

# Set the threshold to debug mode
log4j.appender.FILE.Threshold=debug

# Set the append to false, should not overwrite
log4j.appender.FILE.Append=true

# Set the DatePattern
log4j.appender.FILE.DatePattern='.' yyyy-MM-dd-a

# Define the layout for file appender
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.conversionPattern=%m%n
```

If you like to have an XML configuration file, you can generate as mentioned in initial section and add only additional parameters related to *DailyRollingFileAppender*.

You can try [log4j - Sample Program](#) with the above configuration .